

| | | |
|-----------|----------------|----------------|
| EEEEEEEEE | DDDDDDDDDDDDDD | TTTTTTTTTTTTTT |
| EEEEEEEEE | DDDDDDDDDDDDDD | TTTTTTTTTTTTTT |
| EEEEEEEEE | DDDDDDDDDDDDDD | TTTTTTTTTTTTTT |
| EEE | DDD | TTT |
| EEEEEEEEE | DDD | TTT |
| EEEEEEEEE | DDD | TTT |
| EEEEEEEEE | DDD | TTT |
| EEE | DDD | TTT |
| EEEEEEEEE | DDDDDDDDDDDDDD | TTT |
| EEEEEEEEE | DDDDDDDDDDDDDD | TTT |
| EEEEEEEEE | DDDDDDDDDDDDDD | TTT |

FILEID**WRITEDMSG

M 2

EDTS
V04-

WW RRRRRRRR I III EEEEEE DDDDDDDD TTTTTTTT MM MM SSSSSSS GGGGGGGG
WW RRRRRRRR I III EEEEEE DDDDDDDD TTTTTTTT MM MM SSSSSSS GGGGGGGG
WW RR RR I I EE DD DD TT Mmmm Mmmm SS GG
WW RR RR I I EE DD DD TT Mmmm Mmmm SS GG
WW RR RR I I EE DD DD TT MM MM SS GG
WW RR RR I I EE DD DD TT MM MM SS GG
WW RRRRRRRR I I EEEEEE DD DD TT MM MM SSSSSS GG
WW RRRRRRRR I I EEEEEE DD DD TT MM MM SSSSSS GG
WW WW RR RR I I EE DD DD TT MM MM SS GG GGGGGG
WW WW RR RR I I EE DD DD TT MM MM SS GG GGGGGG
WWWW WWWW RR RR I I EE DD DD TT MM MM SS GG GG
WWWW WWWW RR RR I I EE DD DD TT MM MM SS GG GG
WW WW RR RR I III EEEEEE DDDDDDDD TT MM MM SSSSSSS GGGGGG
WW WW RR RR I III EEEEEE DDDDDDDD TT MM MM SSSSSSS GGGGGG

LL I III SSSSSSSS
LL I III SSSSSSSS
LL I I SS SSSSSS
LLLLLLLLLL I III SSSSSSSS
LLLLLLLLLL I III SSSSSSSS

```
0001 0 XTITLE 'EDTSWRIEDTMSG - write VMSMSG.MSG'
0002 0 MODULE EDTSWRIEDTMSG (
0003 0 IDENT = 'VO4-000',
0004 0 MAIN = EDTSWRIEDTMSG
0005 0 )
0006 1 BEGIN
0007 1
0008 1 ****
0009 1 *
0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0012 1 * ALL RIGHTS RESERVED.
0013 1 *
0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0019 1 * TRANSFERRED.
0020 1 *
0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0023 1 * CORPORATION.
0024 1 *
0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0027 1 *
0028 1 *
0029 1 ****
0030 1 .
0031 1 .
0032 1 ++
0033 1 FACILITY: EDT -- The DEC Standard Editor
0034 1
0035 1 ABSTRACT:
0036 1
0037 1 This module, WRIEDTMSG.FOR, is a FORTRAN program that writes
0038 1 the file VMSMSG.MSG, which is read by the message compiler
0039 1 to produce EDT's run-time messages.
0040 1
0041 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0042 1
0043 1 AUTHOR: John Sauter, CREATION DATE: 23-Jul-1981
0044 1
0045 1 MODIFIED BY:
0046 1
0047 1 1-001 - Original, from BASMSG.FOR, created November 3, 1978, last
0048 1 revised September 24, 1979 (version 1-015). JBS 28-Jul-1981
0049 1 1-002 - Don't omit the first message. JBS 03-Aug-1981
0050 1 1-003 - Change output file name to VMSMSG.MSG. JBS 03-Aug-1981
0051 1 1-004 - Fix output file's module name JBS 06-Aug-1981
0052 1 1-005 - Recoded in BLISS since VMS doesn't like its components to be
0053 1 dependent upon Fortran. JBS 22-Oct-1981
0054 1 1-006 - Change output file name to VMSMSG.TMP. BLS 6-May-1983
0055 1 1-007 - Correct the module header and trailer. JBS 09-May-1983
0056 1 --
0057 1
```

```
59      0058 1 ZSBTTL 'Declarations'  
60      0059 1  
61      0060 1 SWITCHES:  
62      0061 1  
63      0062 1  
64      0063 1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);  
65      0064 1  
66      0065 1  
67      0066 1 LINKAGES:  
68      0067 1  
69      0068 1     NONE  
70      0069 1  
71      0070 1 TABLE OF CONTENTS:  
72      0071 1  
73      0072 1  
74      0073 1 FORWARD ROUTINE  
75      0074 1     EDT$WRIEDTMSG,  
76      0075 1     WRITE_FILE,  
77      0076 1     PRINT,  
78      0077 1     HEX_TEXT,  
79      0078 1     PRINTABLE_TEXT:  
80      0079 1  
81      0080 1  
82      0081 1 INCLUDE FILES:  
83      0082 1  
84      0083 1  
85      0084 1 REQUIRE 'EDTSRC:PSECTS.REQ':  
86      0189 1  
87      0190 1 REQUIRE 'EDTSRC:SYSSYM.REQ':  
88      0220 1  
89      0221 1  
90      0222 1 MACROS:  
91      0223 1  
92      0224 1     NONE  
93      0225 1  
94      0226 1 EQUATED SYMBOLS:  
95      0227 1  
96      0228 1  
97      0229 1 LITERAL  
98      0230 1     EDT$K_FAC_NO = 133;  
99      0231 1  
100     0232 1  
101     0233 1 FIELDS:  
102     0234 1  
103     0235 1     NONE  
104     0236 1  
105     0237 1 STRUCTURES:  
106     0238 1  
107     0239 1     NONE  
108     0240 1  
109     0241 1 PSECTS:  
110     0242 1  
111     0243 1 DECLARE_PSECTS (EDT):  
112     0244 1  
113     0245 1 OWN STORAGE:  
114     0246 1  
115     0247 1     NONE
```

EDT\$WRIEDTMSG
VO4-000 EDT\$WRIEDTMSG - write VMSMSG.MSG
Declarations

C 3
16-Sep-1984 02:18:31 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:25:55 [EDT.SRC]WRIEDTMSG.B32;1

Page 3
(2)

116 0248 1 |
117 0249 1 | EXTERNAL REFERENCES:
118 0250 1 |
119 0251 1 |
120 0252 1 EXTERNAL ROUTINE
121 0253 1 STR\$COPY DX,
122 0254 1 STR\$CONCAT
123 0255 1 LIB\$GET INPUT,
124 0256 1 STR\$COPY R,
125 0257 1 STR\$FREET DX,
126 0258 1 EDT\$MSGTXT;
127 0259 1 | Copy a string, by descriptor
 | Concatenate strings
 | Get a line from SYSSINPUT
 | Copy a string, by reference
 | Free a dynamic string
 | Return the text of a message

EDT\$
VO4-

```
129      0260 1 XSBTTL 'Package of macros for string processing'
130      0261 1 +
131      0262 1 Macro to initialize a dynamic descriptor.
132      0263 1 -
133      0264 1
134      0265 1 MACRO
135      M 0266 1   INIT_DESCRIPTOR (DESCR) =
136      M 0267 1     DESCRIPTOR [DSC$W_LENGTH] = 0;
137      M 0268 1     DESCRIPTOR [DSC$B_DTYPE] = DSC$K_DTYPE_T;
138      M 0269 1     DESCRIPTOR [DSC$B_CLASS] = DSC$K_CLASS_D;
139      M 0270 1     DESCRIPTOR [DSC$A_POINTER] = 0;
140      0271 1 %
141      0272 1 <BLF/MACRO>
142      0273 1 +
143      0274 1 Macro to discard a dynamic descriptor.
144      0275 1 -
145      M 0276 1   DISCARD_DESCRIPTOR (DESCR) =
146      M 0277 1     BEGIN
147      M 0278 1
148      M 0279 1     LOCAL
149      M 0280 1       FREE_STATUS;
150      M 0281 1
151      M 0282 1     FREE_STATUS = STR$FREE1_DX (DESCR);
152      M 0283 1
153      M 0284 1     IF ( NOT .FREE_STATUS) THEN SIGNAL_STOP (.FREE_STATUS);
154      M 0285 1
155      M 0286 1     END;
156      0287 1 %
157      0288 1 +
158      0289 1 Macro to build a text line using FAO. This is a convenience macro.
159      0290 1 -
160      M 0291 1   BUILD_TEXT_LINE (DESCR, CTL_STRING, FAO_ARGS) =
161      M 0292 1     BEGIN
162      M 0293 1
163      M 0294 1     LOCAL
164      M 0295 1       FAO_STATUS,
165      M 0296 1       COPY_STATUS;
166      M 0297 1
167      M 0298 1       CTL_STR_DSC [DSC$W_LENGTH] = %CHARCOUNT (CTL_STRING);
168      M 0299 1       CTL_STR_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
169      M 0300 1       CTL_STR_DSC [DSC$B_CLASS] = DSC$K_CLASS_S;
170      M 0301 1       CTL_STR_DSC [DSC$A_POINTER] = CHSPTR (UPLIT (CTL_STRING));
171      M 0302 1       FAO_STATUS = $FAO (
172      M 0303 1         CTL_STR_DSC,
173      M 0304 1         OUT_LENGTH,
174      M 0305 1         TEMP_STR_DSC,
175      M 0306 1         %REMOVE (FAO_ARGS));
176      M 0307 1
177      M 0308 1     IF ( NOT .FAO_STATUS) THEN SIGNAL_STOP (.FAO_STATUS);
178      M 0309 1
179      M 0310 1     COPY_STATUS = STR$COPY_R (DESCR, OUT_LENGTH, .TEMP_STR_DSC [DSC$A_POINTER]);
180      M 0311 1     .COPY_STATUS
181      M 0312 1     END
182      0313 1 %
183      0314 1 +
184      0315 1 Macro to format and print a line. Errors are returned to the caller.
185      0316 1 This is a convenience macro.
```

EDT\$WRIEDTMSG
V04-000

EDT\$WRIEDTMSG - write VMSMSG.MSG
Package of macros for string processing

E 3
16-Sep-1984 02:18:31
14-Sep-1984 12:25:55
VAX-11 Bliss-32 v4.0-742
[EDT.SRC]WRIEDTMSG.B32;1

Page 5
(3)

EDT
V04

```
186      0317 1 !-
187      M 0318 1 PRINT LINE (TEXT, VARS) =
188      M 0319 1 BEGIN
189      M 0320 1
190      M 0321 1 LOCAL
191      M 0322 1     BUILD_STATUS,
192      M 0323 1     PRINT_STATUS;
193      M 0324 1
194      M 0325 1     BUILD_STATUS = BUILD_TEXT_LINE (LINE_DESC, %STRING (%REMOVE (TEXT)), VARS);
195      M 0326 1
196      M 0327 1     IF ( NOT .BUILD_STATUS) THEN RETURN (.BUILD_STATUS);
197      M 0328 1
198      M 0329 1     PRINT_STATUS = PRINT (.OUTPUT_RAB, LINE_DESC);
199      M 0330 1
200      M 0331 1     IF ( NOT .PRINT_STATUS) THEN RETURN (.PRINT_STATUS);
201      M 0332 1
202      M 0333 1     END
203      M 0334 1
204      M 0335 1     X;
```

206 0336 1 %SBTTL 'EDT\$WRIEDTMSG - Write VMSMSG.TMP'
207 0337 1 ROUTINE EDT\$WRIEDTMSG ! Write VMSMSG.TMP
208 0338 1 =
209 0339 1
210 0340 1 ++
211 0341 1 FUNCTIONAL DESCRIPTION:
212 0342 1
213 0343 1 This routine writes the file VMSMSG.TMP.
214 0344 1
215 0345 1 CALLING SEQUENCE:
216 0346 1
217 0347 1 ret_status.wlc.v = EDT\$WRIEDTMSG ()
218 0348 1
219 0349 1 FORMAL PARAMETERS:
220 0350 1
221 0351 1 NONE
222 0352 1
223 0353 1 IMPLICIT INPUTS:
224 0354 1
225 0355 1 NONE
226 0356 1
227 0357 1 IMPLICIT OUTPUTS:
228 0358 1
229 0359 1 NONE
230 0360 1
231 0361 1 COMPLETION STATUS:
232 0362 1
233 0363 1 SSS_NORMAL Normal successful completion
234 0364 1 Any error from LIB\$GET_INPUT or STR\$FREE1_DX
235 0365 1
236 0366 1 SIDE EFFECTS:
237 0367 1
238 0368 1 Writes a file.
239 0369 1 Any errors from RMSSCREATE, RMSSOPEN, RMSSCONNECT or RMSSCLOSE
240 0370 1 are signalled.
241 0371 1
242 0372 1 --
243 0373 1
244 0374 2 BEGIN
245 0375 2
246 0376 2 LOCAL
247 0377 2 OUTPUT_BUFFER : BLOCK [132, BYTE], ! output buffer, for RMS
248 0378 2 OUTPUT_FAB : \$FAB_DECL, ! RMS FAB for the output file
249 0379 2 OUTPUT_NAM : \$NAM_DECL, ! RMS NAM for the output file
250 0380 2 OUTPLT_RAB : \$RAB_DECL, ! RMS RAB for the output file
251 0381 2 OUTPUT_FILE_NAME_DESC : BLOCK [8, BYTE], ! Name of output file
252 0382 2 OUTPUT_RESULT_NAME : BLOCK [NAMS\$_MAXRSS, BYTE]; ! Place to store output file name
253 0383 2
254 0384 2
255 0385 2 OUTPUT_FILE_NAME_DESC [DSC\$W_LENGTH] = %CHARCOUNT ('VMSMSG');
256 0386 2 OUTPUT_FILE_NAME_DESC [DSC\$B_DTYPE] = DSC\$K_DTYPE_T;
257 0387 2 OUTPUT_FILE_NAME_DESC [DSC\$B_CLASS] = DSC\$K_CLASS_S;
258 0388 2 OUTPUT_FILE_NAME_DESC [DSC\$A_POINTER] = UPLIT ('VMSMSG');
259 0389 2
260 0390 2 + Initialize the FAB, NAM and RAB for the output file
261 0391 2 -
262 P 0392 2 SFAB_INIT (FAB = OUTPUT_FAB, !

```

263 P 0393 2 FAC = (PUT),
264 P 0394 2 FOP = (OFP, SQQ, DFW),
265 P 0395 2 ORG = SEQ,
266 P 0396 2 SHR = NIL,
267 P 0397 2 MRS = 132,
268 P 0398 2 RAT = CR,
269 P 0399 2 RFM = VAR,
270 P 0400 2 FNA = .OUTPUT_FILE_NAME_DESC [DSCSA_POINTER],
271 P 0401 2 FNS = .OUTPUT_FILE_NAME_DESC [DSCSW_LENGTH],
272 P 0402 2 DNA = UPLIT ('EDTSRC:.TMP')
273 P 0403 2 DNS = %CHARCOUNT ('EDTSRC:.fMP'),
274 P 0404 2 NAM = OUTPUT_NAM;
275 P 0405 2 $NAM_INIT (NAM = OUTPUT_NAM,
276 P 0406 2 RSA = OUTPUT_RESULT_NAME,
277 P 0407 2 RSS = NAMSC_MAXRSS);
278 P 0408 2 $RAB_INIT (RAB = OUTPUT_RAB,
279 P 0409 2 RAC = SEQ,
280 P 0410 2 ROP = WBH,
281 P 0411 2 USZ = 132,
282 P 0412 2 UBF = OUTPUT_BUFFER,
283 P 0413 2 FAB = OUTPUT_FAB);

284 0414
285 0415 1+ Create the output file, and do the $CONNECT.
286 0416 1-
287 0417 BEGIN
288 0418
289 0419 LOCAL
290 0420   STATUS,
291 0421   T_STATUS;
292 0422
293 0423 CREATE_STATUS = $CREATE (FAB = OUTPUT_FAB);
294 0424
295 0425 IF ( NOT .CREATE_STATUS)
296 0426 THEN
297 0427   SIGNAL_STOP (
298 0428     SHRS_OPENOUT + (EDTSK_FAC_NO*65536) + STSSK_SEVERE.
299 0429     1,
300 0430     OUTPUT_FILE_NAME_DESC,
301 0431     .OUTPUT_FAB[FABSL_ST$], .OUTPUT_FAB[FABSL_STV]);
302 0432
303 0433 CONNECT_STATUS = $CONNECT (RAB = OUTPUT_RAB);
304 0434
305 0435 IF ( NOT .CONNECT_STATUS)
306 0436 THEN
307 0437   SIGNAL_STOP (
308 0438     SHRS_OPENOUT + (EDTSK_FAC_NO*65536) + STSSK_SEVERE.
309 0439     1,
310 0440     OUTPUT_FILE_NAME_DESC,
311 0441     .OUTPUT_RAB[RABSL_ST$], .OUTPUT_RAB[RABSL_STV]);
312 0442
313 0443 END;
314
315 0444 1+
316 0445 Point the file name descriptor to the resultant name string.
317 0446 1-
318 0447   OUTPUT_FILE_NAME_DESC [DSCSW_LENGTH] = .OUTPUT_NAM [NAMSB_RSL];
319 0448   OUTPUT_FILE_NAME_DESC [DSCSB_DTYPE] = DSCSK_DTPE_T;
            OUTPUT_FILE_NAME_DESC [DSCSB_CLASS] = DSCSK_CLASS_S;

```

EDT\$WRIEDTMSG
V04-000

EDT\$WRIEDTMSG - write VMSMSG.MSG
EDT\$WRIEDTMSG - Write VMSMSG.TMP

H 3
16-Sep-1984 02:18:31 VAX-11 Bliss-32 v4.0-742
14-Sep-1984 12:25:55 [EDT.SRC]WRIEDTMSG.B32;1

Page 8
(4)

EDT
V04-

```
320 0450 2 OUTPUT_FILE_NAME_DESC [DSC$A_POINTER] = .OUTPUT_NAM [NAMSL_RSA];  
321 0451  
322 0452 IF ( NOT WRITE_FILE (OUTPUT_RAB))  
323 0453 THEN SIGNAL_STOP ( !  
324 0454 SHRS_WRITEERR + (EDTSK_FAC_NO*65536) + STSSK_SEVERE,  
325 0455 1,  
326 0456 OUTPUT_FILE_NAME_DESC,  
327 0457 .OUTPUT_RAB [RAB$L_ST$], .OUTPUT_RAB [RAB$L_STV]);  
328 0458  
329 0459  
330 0460 BEGIN  
331 0461 !+ Close the output file.  
332 0462 !-  
333 0463  
334 0464  
335 0465 LOCAL  
336 0466 CLOSE_STATUS;  
337 0467  
338 0468 CLOSE_STATUS = $CLOSE (FAB = OUTPUT_FAB);  
339 0469  
340 0470 IF ( NOT .CLOSE_STATUS)  
341 0471 THEN SIGNAL_STOP ( !  
342 0472 SHRS_CLOSEOUT + (EDTSK_FAC_NO*65536) + STSSK_SEVERE,  
343 0473 1,  
344 0474 OUTPUT_FILE_NAME_DESC,  
345 0475 .OUTPUT_FAB [FAB$L_ST$], .OUTPUT_FAB [FAB$L_STV]);  
346 0476  
347 0477  
348 0478 END;  
349 0479 RETURN (SS$_NORMAL);  
350 0480 1 END;
```

! End of routine EDT\$WRIEDTMSG

.TITLE EDT\$WRIEDTMSG EDT\$WRIEDTMSG - write VMSMSG.MSG
.IDENT \V04-000\

.PSECT _EDT\$CODE,NOWRT, SHR, PIC.2

| | | | | | |
|----------------------------|-------------|--------------|-------------------------|------------------------------------|------|
| 00 50 4D 54 | 00 00 47 53 | 4D 53 4D 56 | 00000 P.AAA: | .ASCII \VMSMSG\<0><0> | 0337 |
| 2E 3A 43 52 | 53 54 44 45 | 00008 P.AAB: | .ASCII \EDTSRC:.TMP\<0> | | |
| | | | | .EXTRN STR\$COPY DX, STR\$CONCAT | |
| | | | | .EXTRN LIB\$GET INPUT, STR\$COPY R | |
| | | | | .EXTRN STR\$FREE1 DX, EDTSMSGTXT | |
| | | | | .EXTRN SYSSCREATE, SYSSCONNECT | |
| | | | | .EXTRN SYSSCLOSE | |
| 007C 00000 EDT\$WRIEDTMSG: | | | | | |
| 0050 8F | 00 | 56 0000000G | 00 9E 00002 | .WORD Save R2,R3,R4,R5,R6 | |
| | | 5E FD80 | CE 9E 00009 | LIB\$STOP, R6 | |
| | 0100 | CE 010E0006 | 8F D0 0000E | -640(SP), SP | |
| | 0104 | CE D2 | AF 9E 00017 | #17694726, OUTPUT_FILE_NAME_DESC | 0385 |
| | | 6E | 00 2C 0001D | P.AAA, OUTPUT_FILE_NAME_DESC+4 | 0388 |
| | | FF2C | CD 00024 | #0, (SP), #0, #80, \$RMS_PTR | 0404 |
| | FF30 | CD 5003 | BF B0 00027 | MOVW #20483, \$RMS_PTR | |
| | | CD 20000060 | 8F D0 0002E | MOVL #536871008, \$RMS_PTR+4 | |

EDTSWRIEDTMSG
VO4-000

EDTSWRIEDTMSG - write VMSMSG.MSG
EDTSWRIEDTMSG - Write VMSMSG.TMP

I 3
16-Sep-1984 02:18:31 VAX-11 BLiss-32 V4.0-742
14-Sep-1984 12:25:55 [EDT.SRC]WRIEDTMSG.B32;1

Page 9
(4)

EDT1
VO4

| | | | | | | | | | | |
|------|----|----|-----------|----|----------|------|-------|-------|--------------------------------|---------------------------------------|
| | | | FF42 | CD | 2001 | 8F | B0 | 00037 | MOVW | #8193, SRMS PTR+22 |
| | | | FF49 | CD | 0200 | 8F | B0 | 0003E | MOVW | #512, SRMS PTR+29 |
| | | | FF4B | CD | FECC | CD | 9E | 0004A | MOVB | #2, \$RMS PTR+31 |
| | | | FF54 | CD | 0104 | CE | D0 | 00051 | MOVAB | OUTPUT_NAM, SRMS_PTR+40 |
| | | | FF58 | CD | 99 | AF | 9E | 00058 | MOVL | OUTPUT_FILE_NAME_DESC+4, SRMS_PTR+44 |
| | | | FF5C | CD | 0100 | CE | 90 | 0005E | MOVAB | P.AAB, SRMS_PTR+48 |
| | | | FF60 | CD | 84 | OB | 90 | 00065 | MOVB | OUTPUTFILE_NAME_DESC, SRMS_PTR+52 |
| | | | FF61 | CD | 00 | 8F | 9B | 0006A | MOVZBW | #11, SRMS_PTR+53 |
| | | | FF62 | CD | 6E | FECC | CD | 00070 | MOVCS | #132, SRMS_PTR+54 |
| | | | | | | | | | | #0, (SP), #0, #96, SRMS_PTR |
| 0060 | 8F | 00 | | | 6002 | 8F | B0 | 0007A | MOVW | #24578, SRMS_PTR |
| | | | FECC | CD | 01 | 8E | 00081 | MNEG8 | #1, SRMS_PTR+2 | |
| 0044 | 8F | 00 | FECE | CD | 6E | 9E | 00086 | MOVAB | OUTPUT_RESULT_NAME, SRMS_PTR+4 | |
| | | | FED0 | CD | 00 | 2C | 0008B | MOVCS | #0, (SP), #0, #68, SRMS_PTR | |
| | | | | | 0108 | CE | 00092 | | | |
| | | | 0108 | CE | 4401 | 8F | B0 | 00095 | MOVW | #17409, SRMS_PTR |
| | | | 010C | CE | 0400 | 8F | 3C | 0009C | MOVZWL | #1024, SRMS_PTR+4 |
| | | | | | 0126 | CE | 94 | 000A3 | CLRB | \$RMS_PTR+30 |
| | | | 0128 | CE | 84 | 8F | 9B | 000A7 | MOVZBW | #132, SRMS_PTR+32 |
| | | | 012C | CE | FF7C | CD | 9E | 000AD | MOVAB | OUTPUT_BUFFER, SRMS_PTR+36 |
| | | | FEC4 | CD | FF2C | CD | 9E | 000B4 | MOVAB | OUTPUT_FAB, SRMS_PTR+60 |
| | | | | | FF2C | CD | 9F | 000BB | PUSHAB | OUTPUT_FAB |
| | | | 00000000G | 00 | | 01 | FB | 000BF | CALLS | #1, SYSSCREATE |
| | | | | 14 | 50 | E8 | 000C6 | BLBS | CREATE_STATUS, 1\$ | |
| | | | | 7E | FF34 | CD | 7D | 000C9 | MOVO | OUTPUT_FAB+8, -(SP) |
| | | | | | 0108 | CE | 9F | 000CE | PUSHAB | OUTPUT_FILE_NAME_DESC |
| | | | | | | 01 | DD | 000D2 | PUSHL | #1 |
| | | | | | 008510A4 | 8F | DD | 000D4 | PUSHL | #8720548 |
| | | | | 66 | 05 | FB | 000DA | CALLS | #5, LIB\$STOP | |
| | | | 00000000G | 00 | 0108 | CE | 9F | 000DD | PUSHAB | OUTPUT_RAB |
| | | | | 17 | | 01 | FB | 000E1 | CALLS | #1, SYSSCONNECT |
| | | | | | 50 | E8 | 000E8 | BLBS | CONNECT_STATUS, 2\$ | |
| | | | | | 0114 | CE | DD | 000EB | PUSHL | OUTPUT_RAB+12 |
| | | | | | 0114 | CE | DD | 000EF | PUSHL | OUTPUT_RAB+8 |
| | | | | | 0108 | CE | 9F | 000F3 | PUSHAB | OUTPUT_FILE_NAME_DESC |
| | | | | | | 01 | DD | 000F7 | PUSHL | #1 |
| | | | | | 008510A4 | 8F | DD | 000F9 | PUSHL | #8720548 |
| | | | | 66 | 05 | FB | 000FF | CALLS | #5, LIB\$STOP | |
| | | | 0100 | CE | FECF | CD | 98 | 00102 | MOVZBW | OUTPUT_NAM+3, OUTPUT_FILE_NAME_DESC |
| | | | 0102 | CE | 010E | 8F | B0 | 00109 | MOVW | #270, OUTPUT_FILE_NAME_DESC+2 |
| | | | 0104 | CE | FED0 | CD | D0 | 00110 | MOVL | OUTPUT_NAM+4, OUTPUT_FILE_NAME_DESC+4 |
| | | | | | 0108 | CE | 9F | 00117 | PUSHAB | OUTPUT_RAB |
| | | | 0000V | CF | | 01 | FB | 00118 | CALLS | #1, WRITE_FILE |
| | | | | 17 | 50 | E8 | 00120 | BLBS | R0, 3\$ | |
| | | | | | 0114 | CE | DD | 00123 | PUSHL | OUTPUT_RAB+12 |
| | | | | | 0114 | CE | DD | 00127 | PUSHL | OUTPUT_RAB+8 |
| | | | | | 0108 | CE | 9F | 0012B | PUSHAB | OUTPUT_FILE_NAME_DESC |
| | | | | | | 01 | DD | 0012F | PUSHL | #1 |
| | | | | | 008510D4 | 8F | DD | 00131 | PUSHL | #8720596 |
| | | | | 66 | 05 | FB | 00137 | CALLS | #5, LIB\$STOP | |
| | | | 00000000G | 00 | FF2C | CD | 9F | 0013A | PUSHAB | OUTPUT_FAB |
| | | | | 14 | 01 | FB | 0013E | CALLS | #1, SYSSCLOSE | |
| | | | | 7E | 50 | E8 | 00145 | BLBS | CLOSE_STATUS, 4\$ | |
| | | | | | FF34 | CD | 7D | 00148 | MOVO | OUTPUT_FAB+8, -(SP) |
| | | | | | 0108 | CE | 9F | 0014D | PUSHAB | OUTPUT_FILE_NAME_DESC |
| | | | | | | 01 | DD | 00151 | PUSHL | #1 |

EDTSWRIEDTMSG
VO4-000

EDTSWRIEDTMSG - write VMSMSG.MSG
EDTSWRIEDTMSG - Write VMSMSG.TMP

J 3
16-Sep-1984 02:18:31 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:25:55 [EDT.SRC]WRIEDTMSG.B32;1

Page 10
(4)

EDTS
VO4-

0085105C 8F DD 00153
66 05 FB 00159
50 01 D0 0015C 48:
04 0015F

PUSHL #8720476
CALLS #5, LIB\$STOP
MOVL #1, R0
RET

; 0473
; 0479
; 0480

; Routine Size: 352 bytes. Routine Base: _EDT\$CODE + 0014

; Rc

```
352 0481 1 %SBTTL 'WRITE_FILE - Actually write the file'  
353 0482 1 ROUTINE WRITE_FILE (  
354 0483 1     OUTPUT_RAB  
355 0484 1 ) =  
356 0485 1 ++  
357 0486 1 FUNCTIONAL DESCRIPTION:  
358 0487 1 This routine writes each record on the specified RAB.  
359 0488 1  
360 0489 1 CALLING SEQUENCE:  
361 0490 1  
362 0491 1 ret_status.wlc.v = WRITE_FILE (OUTPUT_RAB.mz.r)  
363 0492 1  
364 0493 1 FORMAL PARAMETERS:  
365 0494 1  
366 0495 1     OUTPUT_RAB           RAB onto which to write the text  
367 0496 1  
368 0497 1  
369 0498 1  
370 0499 1 IMPLICIT INPUTS:  
371 0500 1  
372 0501 1     NONE  
373 0502 1  
374 0503 1 IMPLICIT OUTPUTS:  
375 0504 1  
376 0505 1     None  
377 0506 1  
378 0507 1 COMPLETION STATUS:  
379 0508 1  
380 0509 1     SSS_NORMAL      Normal successful completion  
381 0510 1     Any errors from RMS $PUT  
382 0511 1  
383 0512 1 SIDE EFFECTS:  
384 0513 1  
385 0514 1     Writes on the file connected to OUTPUT_RAB  
386 0515 1  
387 0516 1 --  
388 0517 1  
389 0518 2 BEGIN  
390 0519 2  
391 0520 2 MAP  
392 0521 2     OUTPUT_RAB : REF SRAB_DECL;  
393 0522 2  
394 0523 2 LOCAL  
395 0524 2  
396 0525 2 * Stuff for BUILD_TEXT_LINE  
397 0526 2  
398 0527 2     CTL_STR DSC : BLOCK [8, BYTE]  
399 0528 2     TEMP_STR DSC : BLOCK [8, BYTE]  
400 0529 2     TEMP_STRING : VECTOR [132, BYTE],  
401 0530 2     OUT_LENGTH,  
402 0531 2  
403 0532 2 * Stuff for PRINT_LINE  
404 0533 2  
405 0534 2     LINE_DESC : BLOCK [8, BYTE],  
406 0535 2  
407 0536 2 * End of stuff for PRINT_LINE  
408 0537 2
```

```

409      0538 2 PRINTABLE_DESC : BLOCK [8, BYTE];
410      0539 2 HEX_DESC : BLOCK [8, BYTE];
411      0540 2 TEXT_DESC : BLOCK [8, BYTE];
412      0541 2 NAME_DESC : BLOCK [8, BYTE];
413      0542 2 SEVERITY_DESC : BLOCK [8, BYTE];
414      0543 2 NAME_LENGTH,
415      0544 2 TEXT_LENGTH,
416      0545 2 SEVERITY_LENGTH,
417      0546 2 SEVERITY_ADDR : REF VECTOR [, BYTE];

!+ Set up TEMP_STR_DSC for BUILD_TEXT_LINE
!-
420      0548 2
421      0549 2 TEMP_STR_DSC [DSC$W_LENGTH] = 132;
422      0550 2 TEMP_STR_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
423      0551 2 TEMP_STR_DSC [DSC$B_CLASS] = DSC$K_CLASS_S;
424      0552 2 TEMP_STR_DSC [DSC$A_POINTER] = CHSPTR (TEMP_STRING);

!+ Set up LINE_DESC for PRINT_LINE, etc.
!-
425      0553 2
426      0554 2 INIT_DESCRIPTOR (LINE_DESC);
427      0555 2 INIT_DESCRIPTOR (PRINTABLE_DESC);
428      0556 2 INIT_DESCRIPTOR (HEX_DESC);
429      0557 2 INIT_DESCRIPTOR (TEXT_DESC);
430      0558 2 INIT_DESCRIPTOR (NAME_DESC);
431      0559 2 INIT_DESCRIPTOR (SEVERITY_DESC);

!+ Put out the initial information.
!-
432      0560 2
433      0561 2 PRINT_LINE (<'!!! This file, VMSMSG.TMP, contains the definitions of the EDT'>, <' '>);
434      0562 2 PRINT_LINE (<'!!! messages for VAX/VMS. This file is read by the MESSAGE compiler'>, <' '>);
435      0563 2 PRINT_LINE (<'!!! to build an object file containing the EDT messages.'>, <' '>);
436      0564 2 PRINT_LINE (<'!!!'>, <' '>);
437      0565 2 PRINT_LINE (<'!!! .TITLE EDTSVMSMSG EDT''s message text'>, <' '>);
438      0566 2 PRINT_LINE (<'!!!'>, <' '>);
439      0567 2 PRINT_LINE (<'!.FACILITY/SYSTEM EDT, !SL'>, <EDTSK_FAC_NO>);

!+ Write a line for each message
!-
440      0575 2
441      0576 2
442      0577 2
443      0578 2 INCR CODE FROM 0 TO 4095 DO
444      0579 2 BEGIN
445      0580 2   EDTSMSGTXT (CODE, SEVERITY_DESC, NAME_LENGTH, NAME_DESC, TEXT_LENGTH, TEXT_DESC);
446      0581 2
447      0582 2 If the severity field is blank we are done.
448      0583 2
449      0584 2   SEVERITY_ADDR = .SEVERITY_DESC [DSC$A_POINTER];
450      0585 2
451      0586 2   IF (.SEVERITY_ADDR [0] EQL ' ') THEN EXITLOOP;
452      0587 2
453      0588 2   PRINT_LINE (<'!AS/!AS <!AS>'>, <NAME_DESC, SEVERITY_DESC, TEXT_DESC>);
454      0589 2   END;

!+ Write out the trailer line
!-
455      0590 2
456      0591 2
457      0592 2
458      0593 2
459      0594 2   PRINT_LINE (<'!.END'>, <' '>);

```

**EDTSWRIEDTMSG
V04-000**

EDTSWRIEDTMSG - write VMSMSG.MSG
WRITE FILE - Actually write the file

M 3
16-Sep-1994 02:18:31 VAX-11 Bliss-32 v4.0-742
14-Sep-1984 12:25:55 [EDT.SRC]WRIEDTMSG.B32;1

Page 13
(5)

EDTS
V04-

! End of routine WRITE_FILE

```

56 20 2C 65 6C 69 66 20 73 69 68 54 2E 6F 50 4D 54 20 65 68 74 20 73 69 6E 53 4D 53 20 6E 69 61 4D 00174 P.AAC: .ASCII \!! This file, VMSMSG.TMP, contains the d\
74 6E 6F 63 20 2C 64 20 6E 6F 69 74 69 6F 65 40 00 00 54 44 45 6E 69 66 65 68 00183 .ASCII \00192
74 20 66 6F 20 73 65 67 61 73 73 65 6D 60 2F 58 41 56 20 6C 69 66 20 0019C .ASCII \efinitions of the EDT\<0><0><0>
72 6F 66 20 73 65 67 61 73 73 65 6D 60 2F 58 41 56 20 6C 69 66 20 001AB P.AAD: .ASCII \!! messages for VAX/VMS. This file is r\
73 69 68 54 20 20 72 20 73 69 20 65 6C 69 6D 63 20 6F 63 20 6F 63 65 6A 62 6F 001C3 .ASCII \ead by the MESSAGE compiler\<0>
53 53 45 4D 20 65 68 74 20 79 62 20 64 61 65 00 47 41 001DC .ASCII \ead by the MESSAGE compiler\<0>
00 72 65 6C 69 70 6D 6F 63 20 6F 63 20 6F 63 65 6A 62 6F 001EB .ASCII \ead by the MESSAGE compiler\<0>
20 6E 61 20 64 6C 69 75 62 20 6F 74 63 65 6A 62 6F 001F8 P.AAE: .ASCII \!! to build an object file containing th\
6E 6F 63 20 65 6C 69 66 20 74 67 6E 69 6E 69 61 74 00207 .ASCII \ead by the MESSAGE compiler\<0>
2E 73 65 67 61 73 73 65 6D 20 54 44 45 20 65 00220 .ASCII \e EDT messages.\<0>
00 0022F .ASCII \ead by the MESSAGE compiler\<0>
20 45 4C 54 49 54 2E 20 20 20 20 20 00 00 21 21 00230 P.AAF: .ASCII \!!\<0><0>
27 54 44 45 20 47 53 40 53 40 56 24 54 44 45 00234 P.AAG: .ASCII \ .TITLE EDT$VMSMSG EDT's message \
20 20 65 67 61 73 73 65 6D 20 74 78 65 74 00243 .ASCII \ead by the MESSAGE compiler\<0>
54 00 44 45 20 47 53 40 53 40 56 24 54 44 45 00252 .ASCII \ead by the MESSAGE compiler\<0>
45 54 53 59 53 2F 59 54 49 4C 49 43 41 46 2E 0025C .ASCII \text\
00 00 00 00 4C 53 21 20 2C 54 44 45 20 4D 00260 P.AAH: .ASCII \!!\<0><0>
45 54 53 59 53 2F 59 54 49 4C 49 43 41 46 2E 00264 P.AAI: .ASCII \.FACILITY/SYSTEM EDT, !SL\<0><0><0>
00 00 00 00 4C 53 21 20 2C 54 44 45 20 4D 00273 .ASCII \.FACILITY/SYSTEM EDT, !SL\<0><0><0>
00 00 00 00 4C 53 21 20 2C 54 44 45 20 4D 00280 P.AAJ: .ASCII \!AS!/AS <!AS>\<0><0><0>
00 00 00 00 4C 53 21 20 2C 54 44 45 20 4D 0028F .ASCII \!AS!/AS <!AS>\<0><0><0>
00 00 00 00 4C 53 21 20 2C 54 44 45 20 4D 00290 P.AAK: .ASCII \.END\
44 4E 45 2E 00290 P.AAK: .ASCII \.END\
.EXTN SYSSFAO

```

01FC 00000 WRITE_FILE:

| | | | | | | |
|----|-------------|----|----------|-------|-----------------------------|---------------------------|
| | | | | | WORD | Save R2,R3,R4,R5,R6,R7,R8 |
| 58 | 0000V | CF | 9E 00002 | MOVAB | PRINT, R8 | |
| 57 | 00000000G | 00 | 9E 00007 | MOVAB | STR\$FREE1 DX, R7 | |
| 56 | 00000000G | 00 | 9E 0000E | MOVAB | STRSCOPY -R R6 | |
| 55 | 00000000G | 00 | 9E 00015 | MOVAB | SYSSFAO, -R5 | |
| 54 | 00000000G | 00 | 9E 0001C | MOVAB | LIBSTOP, R4 | |
| 5E | FF2C | CE | 9E 00023 | MOVAB | -212(SP), SP | |
| F0 | AD 010E0084 | 8F | D0 00028 | MOVL | #17694852, TEMP STR DSC | |
| F4 | AD 40 | AE | 9E 00030 | MOVAB | TEMP STRING, TEMP STR_DSC+4 | |
| 38 | AE 020E0000 | 8F | D0 00035 | MOVL | #34471936, LINE_DESC | |
| | 3C | AE | D4 0003D | CLRL | LINE DESC+4 | |
| 30 | AE 020E0000 | 8F | D0 00040 | MOVL | #34471936, PRINTABLE_DESC | |
| | 34 | AE | D4 00048 | CLRL | PRINTABLE_DESC+4 | |

EDTSWRIEDTMSG
VO4-000EDTSWRIEDTMSG - write VMSMSG.MSG
WRITE_FILE - Actually write the fileN 3
16-Sep-1984 02:18:31
14-Sep-1984 12:25:55
VAX-11 Bliss-32 V4.0-742
[EDT.SPCJWR!EDTMSG.B3?;1]Page 14
(5)EDT
VO4

| | | | | | |
|----|-------------|----|-------------|--------|--------------------------|
| 28 | AE 020E0000 | 8F | D0 0004B | MOVL | #34471936, HEX_DESC |
| | 2C | AE | D4 00053 | CLRL | HEX_DESC+4 |
| 20 | AE 020E0000 | 8F | D0 00056 | MOVL | #34471936, TEXT_DESC |
| | 24 | AE | D4 0005E | CLRL | TEXT_DESC+4 |
| 18 | AE 020E0000 | 8F | D0 00061 | MOVL | #34471936, NAME_DESC |
| | 1C | AE | D4 00069 | CLRL | NAME_DESC+4 |
| 10 | AE 020E0000 | 8F | D0 0006C | MOVL | #34471936, SEVERITY_DESC |
| | 14 | AE | D4 00074 | CLRL | SEVERITY_DESC+4 |
| F8 | AD 010E003D | 8F | D0 00077 | MOVL | #17694787, CTL_STR_DSC |
| FC | AD FESD | CF | 9E 0007F | MOVAB | P_AAC, CTL_STR_DSC+4 |
| | | 20 | DD 00085 | PUSHL | #32 |
| | | F0 | AD 9F 00087 | PUSHAB | TEMP_STR_DSC |
| | | 14 | AE 9F 0008A | PUSHAB | OUT_LENGTH |
| | | F8 | AD 9F 0008D | PUSHAB | CTL_STR_DSC |
| 65 | | 04 | FB 00090 | CALLS | #4, SYSSFAO |
| 05 | | 50 | E8 00093 | BLBS | FAO_STATUS, 1\$ |
| | | 50 | DD 00096 | PUSHL | FAO_STATUS |
| 64 | | 01 | FB 00098 | CALLS | #1, LIBSTOP |
| | | F4 | AD DD 00098 | PUSHL | TEMP_STR_DSC+4 |
| | | 10 | AE 9F 0009E | PUSHAB | OUT_LENGTH |
| | | 40 | AE 9F 000A1 | PUSHAB | LINE_DESC |
| 66 | | 03 | FB 000A4 | CALLS | #3, STRSCOPY_R |
| 7D | | 50 | E9 000A7 | BLBC | BUILD_STATUS, 4\$ |
| | | 38 | AE 9F 000AA | PUSHAB | LINE_DESC |
| 52 | | 04 | AC D0 000AD | MOVL | OUTPUT_RAB, R2 |
| | | 52 | DD 000B1 | PUSHL | R2 |
| 68 | | 02 | FB 000B3 | CALLS | #2, PRINT |
| F8 | AD 010E0043 | 50 | E9 000B6 | BLBC | PRINT_STATUS, 5\$ |
| FC | AD FESB | 8F | D0 000B9 | MOVL | #17694787, CTL_STR_DSC |
| | | CF | 9E 000C1 | MOVAB | P_AAD, CTL_STR_DSC+4 |
| | | 20 | DD 000C7 | PUSHL | #32 |
| | | F0 | AD 9F 000C9 | PUSHAB | TEMP_STR_DSC |
| | | 14 | AE 9F 000CC | PUSHAB | OUT_LENGTH |
| | | F8 | AD 9F 000CF | PUSHAB | CTL_STR_DSC |
| 65 | | 04 | FB 000D2 | CALLS | #4, SYSSFAO |
| 05 | | 50 | E8 000D5 | BLBS | FAO_STATUS, 2\$ |
| | | 50 | DD 000D8 | PUSHL | FAO_STATUS |
| 64 | | 01 | FB 000DA | CALLS | #1, LIBSTOP |
| | | F4 | AD DD 000DD | PUSHL | TEMP_STR_DSC+4 |
| | | 10 | AE 9F 000E0 | PUSHAB | OUT_LENGTH |
| | | 40 | AE 9F 000E3 | PUSHAB | LINE_DESC |
| 66 | | 03 | FB 000E6 | CALLS | #3, STRSCOPY_R |
| 79 | | 50 | E9 000E9 | BLBC | BUILD_STATUS, 7\$ |
| | | 38 | AE 9F 000EC | PUSHAB | LINE_DESC |
| | | 52 | DD 000EF | PUSHL | R2 |
| 68 | | 02 | FB 000F1 | CALLS | #2, PRINT |
| F8 | AD 010E0037 | 50 | E9 000F4 | BLBC | PRINT_STATUS, 8\$ |
| FC | AD FE61 | 8F | D0 000F7 | MOVL | #17694775, CTL_STR_DSC |
| | | CF | 9E 000FF | MOVAB | P_AAE, CTL_STR_DSC+4 |
| | | 20 | DD 00105 | PUSHL | #32 |
| | | F0 | AD 9F 00107 | PUSHAB | TEMP_STR_DSC |
| | | 14 | AE 9F 0010A | PUSHAB | OUT_LENGTH |
| | | F8 | AD 9F 0010D | PUSHAB | CTL_STR_DSC |
| 65 | | 04 | FB 00110 | CALLS | #4, SYSSFAO |
| 05 | | 50 | E8 00113 | BLBS | FAO_STATUS, 3\$ |
| | | 50 | DD 00116 | PUSHL | FAO_STATUS |
| 64 | | 01 | FB 00118 | CALLS | #1, LIBSTOP |

EDT\$WRIEDTMSG
V04-000EDT\$WRIEDTMSG - write VMSMSG.MSG
WRITE_FILE - Actually write the fileB 4
16-Sep-1984 02:18:31
14-Sep-1984 12:25:55
VAX-11 Bliss-32 V4.0-742
[EDT.SRC]WRIEDTMSG.B32;1Page 15
(5)EDT
V04-

| | | | | | | | |
|----|-------------|----|-------------|----------|--------|------------------------|--------------------|
| | | F4 | AD | DD 0011B | 38: | PUSHL | TEMP STR DSC+4 |
| | | 10 | AE | 9F 0011E | | PUSHAB | OUT LENGTH |
| | | 40 | AF | 9F 00121 | | PUSHAB | LINE DESC |
| | 66 | 79 | 03 | FB 00124 | | CALLS | #3, STR\$COPY_R |
| | | 38 | AE | 9F 0012A | 48: | BLBC | BUILD STATUS, 10\$ |
| | | | S2 | DD 0012D | | PUSHAB | LINE_DESC |
| | 68 | 79 | 02 | FB 0012F | | PUSHL | R2 |
| F8 | AD 010E0002 | 50 | E9 00132 | 55: | CALLS | #2, PRINT | |
| FC | AD FE5B | 8F | DO 00135 | | BLBC | PRINT STATUS, 11\$ | |
| | | CF | 9E 0013D | | MOVL | #17694722, CTL_STR_DSC | |
| | | 20 | DD 00143 | | MOVAB | P_AAF, CTL_STR_DSC+4 | |
| | | F0 | AD 9F 00145 | | PUSHL | #32 | |
| | | 14 | AE 9F 00148 | | PUSHAB | TEMP STR DSC | |
| | | F8 | AD 9F 0014B | | PUSHAB | OUT LENGTH | |
| | 65 | 05 | 04 | FB 0014E | | CALLS | CTL_STR DSC |
| | | 50 | E8 00151 | | BLBS | #4, SYSSFAO | |
| | | 50 | DD 00154 | | PUSHL | FAO_STATUS, 6\$ | |
| | 64 | 01 | FB 00156 | | CALLS | FAO_STATUS | |
| F8 | AD 010E002C | F4 | AD DD 00159 | 68: | PUSHL | #1, LIB\$STOP | |
| FC | AD FE21 | 10 | AE 9F 0015C | | PUSHAB | TEMP STR DSC+4 | |
| | | 40 | AE 9F 0015F | | PUSHAB | OUT LENGTH | |
| | 66 | 79 | 03 | FB 00162 | | CALLS | LINE DESC |
| | | 50 | E9 00165 | 78: | BLBC | #3, STR\$COPY_R | |
| | | 38 | AE 9F 00168 | | PUSHAB | BUILD STATUS, 13\$ | |
| | | | S2 | DD 0016B | | PUSHL | LINE_DESC |
| | 68 | 79 | 02 | FB 0016D | | CALLS | R2 |
| F8 | AD 010E002C | 50 | E9 00170 | 85: | BLBC | #2, PRINT | |
| FC | AD FE21 | 8F | DO 00173 | | MOVL | PRINT STATUS, 14\$ | |
| | | CF | 9E 0017B | | MOVAB | #17694764, CTL_STR_DSC | |
| | | 20 | DD 00181 | | PUSHL | P_AAG, CTL_STR_DSC+4 | |
| | | F0 | AD 9F 00183 | | PUSHAB | #32 | |
| | | 14 | AE 9F 00186 | | PUSHAB | TEMP STR DSC | |
| | | F8 | AD 9F 00189 | | PUSHAB | OUT LENGTH | |
| | 65 | 05 | 04 | FB 0018C | | CALLS | CTL_STR DSC |
| | | 50 | E8 0018F | | BLBS | #4, SYSSFAO | |
| | | 50 | DD 00192 | | PUSHL | FAO_STATUS, 9\$ | |
| | 64 | 01 | FB 00194 | | CALLS | FAO_STATUS | |
| F8 | AD 010E0002 | F4 | AD DD 00197 | 95: | PUSHL | #1, LIB\$STOP | |
| FC | AD FEOF | 10 | AE 9F 0019A | | PUSHAB | TEMP STR DSC+4 | |
| | | 40 | AE 9F 0019D | | PUSHAB | OUT LENGTH | |
| | 66 | 78 | 03 | FB 001A0 | | CALLS | LINE DESC |
| | | 50 | E9 001A3 | 105: | BLBC | #3, STR\$COPY_R | |
| | | 38 | AE 9F 001A6 | | PUSHAB | BUILD STATUS, 16\$ | |
| | | | S2 | DD 001A9 | | PUSHL | LINE_DESC |
| | 68 | 78 | 02 | FB 001AB | | CALLS | R2 |
| F8 | AD 010E0002 | 50 | E9 001AE | 115: | BLBC | #2, PRINT | |
| FC | AD FEOF | 8F | DO 001B1 | | MOVL | PRINT STATUS, 17\$ | |
| | | CF | 9E 001B9 | | MOVAB | #17694722, CTL_STR_DSC | |
| | | 20 | DD 001BF | | PUSHL | P_AAH, CTL_STR_DSC+4 | |
| | | F0 | AD 9F 001C1 | | PUSHAB | #32 | |
| | | 14 | AE 9F 001C4 | | PUSHAB | TEMP STR DSC | |
| | | F8 | AD 9F 001C7 | | PUSHAB | OUT LENGTH | |
| | 65 | 05 | 04 | FB 001CA | | CALLS | CTL_STR DSC |
| | | 50 | E8 001CD | | BLBS | #4, SYSSFAO | |
| | | 50 | DD 001D0 | | PUSHL | FAO_STATUS, 12\$ | |
| | 64 | 01 | FB 001D2 | | CALLS | FAO_STATUS | |
| | | | | | | #1, LIB\$STOP | |

0570

0571

0572

Si
Ru
El
Li
Le
Ma
Co

EDTSWRIEDTMSG
V04-000 EDTSWRIEDTMSG - write VMSMSG.MSG
 WRITE_FILE - Actually write the file

C 4
16-Sep-1984 02:18:31 14-Sep-1984 12:25:55 VAX-11 Bliss-32 V4.0-742
[EDT.SRC]WRIEDTMSG.B32:1

Page 16
(5)

**FI

| | | | | | | | | | | |
|----|----|----------|----|----|-------|-------|--------------------|------------------------|--------------------------------|--|
| | | F4 | AD | DD | 001D5 | 12\$: | PUSHL | TEMP STR DSC+4 | | |
| | | 10 | AE | 9F | 001D8 | | PUSHAB | OUT LENGTH | | |
| | | 40 | AE | 9F | 001DB | | PUSHAB | LINE DESC | | |
| | | 66 | 48 | 03 | FB | 001DE | CALLS | #3, STRSCOPY_R | | |
| | | | | 50 | E9 | 001E1 | BLBC | BUILD STATUS, 17\$ | | |
| | | | | 38 | AE | 9F | PUSHAB | LINE_DESC | | |
| | | | | 52 | DD | 001E7 | PUSHL | R2 | | |
| | | | | 68 | 02 | FB | CALLS | #2, PRINT | | |
| | | | | 3D | 50 | E9 | 001E9 | PRINT STATUS, 17\$ | | |
| F8 | AD | 010E0019 | 8F | DO | 001EF | | BLBC | #17694745, CTL_STR_DSC | | |
| FC | AD | FDD5 | CF | 9E | 001F7 | | MOVL | P.AAI, CTL_STR_DSC+4 | | |
| | | | | 7E | 85 | 8F | MOVAB | | | |
| | | | | | F0 | AD | MOVZBL | #133, -(SPT) | | |
| | | | | | 14 | AE | 9F | TEMP STR DSC | | |
| | | | | | F8 | AD | 00201 | OUT LENGTH | | |
| | | | | | 04 | AE | 9F | PUSHAB | | |
| | | | | | 50 | AD | 00204 | PUSHAB | | |
| | | | | | 50 | 50 | DD | PUSHAB | | |
| | | | | | 05 | 50 | 00210 | PUSHL | | |
| | | | | | 65 | 04 | FB | CALLS | | |
| | | | | | 05 | 50 | 0020A | #4, SYSSFAO | | |
| | | | | | 64 | 50 | E8 | BLBS | FAO_STATUS, 15\$ | |
| | | | | | 64 | 01 | DD | FAO_STATUS | | |
| | | | | | | 15\$: | 00210 | #1, LIBSTOP | | |
| | | | | | | | 00212 | TEMP STR DSC+4 | | |
| | | | | | | | 00215 | OUT LENGTH | | |
| | | | | | | | 00218 | PUSHAB | | |
| | | | | | | | 00218 | PUSHAB | | |
| | | | | | | | 00221 | PUSHAB | | |
| | | | | | | | 00221 | PUSHL | | |
| | | | | | | | 00224 | CALLS | | |
| | | | | | | | 00224 | BLBC | #3, STRSCOPY_R | |
| | | | | | | | 00227 | PUSHAB | BUILD STATUS, 20\$ | |
| | | | | | | | 00229 | PUSHL | LINE_DESC | |
| | | | | | | | 0022C | CALLS | R2 | |
| | | | | | | | 0022C | BLBC | #2, PRINT | |
| | | | | | | | 0022F | PUSHAB | PRINT_STATUS, 20\$ | |
| | | | | | | | 0022F | CLRL | CODE | |
| | | | | | | | 00232 | PUSHAB | TEXT_DESC | |
| | | | | | | | 00232 | PUSHAB | TEXT_LENGTH | |
| | | | | | | | 00235 | PUSHAB | NAME_DESC | |
| | | | | | | | 00235 | PUSHAB | NAME_LENGTH | |
| | | | | | | | 00238 | PUSHAB | SEVERITY_DESC | |
| | | | | | | | 00238 | PUSHAB | CODE | |
| | | | | | | | 00241 | CALLS | #6, EDTMSGTXT | |
| | | | | | | | 00244 | MOVL | SEVERITY_DESC+4, SEVERITY_ADDR | |
| | | | | | | | 0024B | CMPB | (SEVERITY_ADDR), #32 | |
| | | | | | | | 0024F | BEQL | 21\$ | |
| | | | | | | | 00252 | MOVL | #17694733, CTL_STR_DSC | |
| | | | | | | | 00254 | MOVAB | P.AAJ, CTL_STR_DSC+4 | |
| F8 | AD | 010E000D | 8F | DO | 00254 | | | | | |
| FC | AD | FD8C | CF | 9E | 0025C | | | | | |
| | | | | | 20 | AE | PUSHAB | | | |
| | | | | | 14 | AE | TEXT_DESC | | | |
| | | | | | 20 | AE | SEVERITY_DESC | | | |
| | | | | | F0 | AD | NAME_DESC | | | |
| | | | | | 1C | AE | NAME_LENGTH | | | |
| | | | | | F8 | AD | OUT LENGTH | | | |
| | | | | | 06 | FB | CTL_STR_DSC | | | |
| | | | | | 05 | 50 | #6, SYSSFAO | | | |
| | | | | | 05 | 50 | FAO_STATUS | | | |
| | | | | | 64 | 50 | 19\$: | | | |
| | | | | | 64 | 01 | CALLS | | | |
| | | | | | 64 | AD | PUSHL | | | |
| | | | | | 64 | AE | TEMP STR DSC+4 | | | |
| | | | | | 64 | AE | OUT LENGTH | | | |
| | | | | | 64 | AE | LINE DESC | | | |
| | | | | | 66 | 03 | #3, STRSCOPY_R | | | |
| | | | | | 66 | 50 | BUILD STATUS, 23\$ | | | |
| | | | | | 66 | AE | LINE_DESC | | | |
| | | | | | 4F | 38 | | | | |

EDTSWRIEDTMSG - write VMSMSG.MSG
VO4-000 **WRITE_FILE** - Actually write the file

D 4
16-Sep-1984 02:18:31 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:25:55 [EDT.SRC]WRIEDTMSG.B32;1

Page 17
(5)

| 90 | 68 | 52 | DD | 00291 | PUSHL | R2 | | |
|----|----|----|-------|-------|--------|------------------------|--|--|
| | 44 | 02 | FB | 00293 | CALLS | #2, PRINT | | |
| | AE | 50 | E9 | 00296 | BLBC | PRINT STATUS, 23\$ | | |
| 08 | F8 | 8F | F3 | 00299 | AOBLEQ | #4095- CODE, 18\$ | | |
| | AD | AD | DO | 002A2 | MOVL | #17694724, CTL_STR_DSC | | |
| FC | AD | CF | 9E | 002AA | MOVAB | P_AAK, CTL_STR_DSC+4 | | |
| | | 20 | DD | 002B0 | PUSHL | #32 | | |
| | | F0 | AD | 9F | PUSHAB | TEMP STR DSC | | |
| | | 14 | AE | 9F | PUSHAB | OUT_LENGTH | | |
| | | F8 | AD | 9F | PUSHAB | CTL_STR DSC | | |
| | 65 | 04 | FB | 002BB | CALLS | #4, SYS\$FAO | | |
| 05 | | 50 | E8 | 002BE | BLBS | FAO_STATUS, 22\$ | | |
| | | 50 | DD | 002C1 | PUSHL | FAO_STATUS | | |
| 64 | | 01 | FB | 002C3 | CALLS | #1, LIB\$STOP | | |
| | | F4 | AD | DD | PUSHL | TEMP_STR_DSC+4 | | |
| | | 10 | AE | 9F | PUSHAB | OUT_LENGTH | | |
| | | 40 | AE | 9F | PUSHAB | LINE_DESC | | |
| 66 | | 03 | FB | 002CF | CALLS | #3, STR\$COPY_R | | |
| 62 | | 50 | E9 | 002D2 | BLBC | BUILD_STATUS, 30\$ | | |
| | | 38 | AE | 9F | PUSHAB | LINE_DESC | | |
| | | 52 | DD | 002D8 | PUSHL | R2 | | |
| 68 | | 02 | FB | 002DA | CALLS | #2, PRINT | | |
| 57 | | 50 | E9 | 002DD | BLBC | PRINT STATUS, 30\$ | | |
| | | 38 | AE | 9F | PUSHAB | LINE_DESC | | |
| 67 | | 01 | FB | 002E0 | CALLS | #1, STR\$FREE1 DX | | |
| 05 | | 50 | E8 | 002E6 | BLBS | FRÉE_STATUS, 24\$ | | |
| | | 50 | DD | 002E9 | PUSHL | FREE_STATUS | | |
| 64 | | 01 | FB | 002EB | CALLS | #1, [IB\$STOP | | |
| | | 30 | AE | 9F | PUSHAB | PRINTABLE_DESC | | |
| 67 | | 01 | FB | 002F1 | CALLS | #1, STR\$FREE1 DX | | |
| 05 | | 50 | E8 | 002F4 | BLBS | FRÉE_STATUS, 25\$ | | |
| | | 50 | DD | 002F7 | PUSHL | FREE_STATUS | | |
| 64 | | 01 | FB | 002F9 | CALLS | #1, [IB\$STOP | | |
| | | 28 | AE | 9F | PUSHAB | HEX_DESC | | |
| 67 | | 01 | FB | 002FF | CALLS | #1, STR\$FREE1 DX | | |
| 05 | | 50 | E8 | 00302 | BLBS | FRÉE_STATUS, 26\$ | | |
| | | 50 | DD | 00305 | PUSHL | FREE_STATUS | | |
| 64 | | 01 | FB | 00307 | CALLS | #1, [IB\$STOP | | |
| | | 20 | AE | 9F | PUSHAB | TEXT_DESC | | |
| 67 | | 01 | FB | 0030D | CALLS | #1, STR\$FREE1 DX | | |
| 05 | | 50 | E8 | 00310 | BLBS | FRÉE_STATUS, 27\$ | | |
| | | 50 | DD | 00313 | PUSHL | FREE_STATUS | | |
| 64 | | 01 | FB | 00315 | CALLS | #1, [IB\$STOP | | |
| | | 18 | AE | 9F | PUSHAB | NAMÉ_DESC | | |
| 67 | | 01 | FB | 0031B | CALLS | #1, STR\$FREE1 DX | | |
| 05 | | 50 | E8 | 0031E | BLBS | FRÉE_STATUS, 28\$ | | |
| | | 50 | DD | 00321 | PUSHL | FREE_STATUS | | |
| 64 | | 01 | FB | 00323 | CALLS | #1, [IB\$STOP | | |
| | | 10 | AE | 9F | PUSHAB | SEVERITY_DESC | | |
| 67 | | 01 | FB | 00329 | CALLS | #1, STR\$FREE1 DX | | |
| 05 | | 50 | E8 | 0032C | BLBS | FRÉE_STATUS, 29\$ | | |
| | | 50 | DD | 0032F | PUSHL | FREE_STATUS | | |
| 64 | | 01 | FB | 00331 | CALLS | #1, [IB\$STOP | | |
| 50 | | 01 | DO | 00334 | MOVL | #1, R0 | | |
| | | 04 | 00337 | 30\$: | RET | | | |

; Routine Size: 824 bytes, Routine Base: _EDT\$CODE + 0294

EDTSURIEDTMSG
VO4-000

EDTSURIEDTMSG - write VM\$MSG.MSG
WRITE_FILE - Actually write the file

E⁴
16-Sep-1984 02:18:31
12-Sep-1984 12:25:55

VAX-11 Bliss-32 V4.0-742
[EDT.SRC]WRIEDTMSG.B32;1

Page 18
(5)

EDT
VO4-

```

478 0606 1 ZSBTTL 'PRINT - print a text line on a file'
479 0607 1 ROUTINE PRINT (
480 0608 1   RAB_ADDR
481 0609 1   TEXT_LINE
482 0610 1   ) =
483 0611 1
484 0612 1 ++
485 0613 1 | FUNCTIONAL DESCRIPTION:
486 0614 1 | This routine interfaces to RMS to print a line of text.
487 0615 1
488 0616 1 | CALLING SEQUENCE:
489 0617 1
490 0618 1
491 0619 1 | ret_status.wlc.v = PRINT (RAB_ADDR.mz.r, TEXT_LINE.rt.dx)
492 0620 1
493 0621 1 | FORMAL PARAMETERS:
494 0622 1
495 0623 1 | RAB_ADDR
496 0624 1 | TEXT_LINE
497 0625 1 | Pointer to the RAB onto which to print the line of text
498 0626 1 | Descriptor for the line of text to be printed.
499 0627 1 | IMPLICIT INPUTS:
500 0628 1 | NONE
501 0629 1
502 0630 1 | IMPLICIT OUTPUTS:
503 0631 1 | NONE
504 0632 1
505 0633 1
506 0634 1 | COMPLETION STATUS:
507 0635 1
508 0636 1 | SSS_NORMAL Normal successful completion
509 0637 1 | ALL RMS errors are returned to the caller, so that they can be
510 0638 1 | signalled with the file name.
511 0639 1
512 0640 1 | SIDE EFFECTS:
513 0641 1
514 0642 1 | Does a $PUT to the RAB.
515 0643 1
516 0644 1 | --
517 0645 1
518 0646 2 | BEGIN
519 0647 2
520 0648 2 | MAP
521 0649 2 |   RAB_ADDR : REF $RAB DECL,
522 0650 2 |   TEXT_LINE : REF BLOCK [8, BYTE];
523 0651 2
524 0652 2 | LOCAL
525 0653 2 |   PUT_STATUS;
526 0654 2
527 0655 2 | +
528 0656 2 | | Fill in the RAB fields.
529 0657 2 | -
530 0658 2 | | RAB_ADDR [RABSU_RSZ] = .TEXT_LINE [DSC$W_LENGTH];
531 0659 2 | | RAB_ADDR [RABSL_RBF] = .TEXT_LINE [DSC$A_POINTER];
532 0660 2 | +
533 0661 2 | | Now do the $PUT
534 0662 2 | -

```

EDTSWRIEDTMSG
VO4-000

EDTSWRIEDTMSG - write VMSMSG.MSG
PRINT - print a text line on a file

G 4
16-Sep-1984 02:18:31
14-Sep-1984 12:25:55

VAX-11 Bliss-32 V4.0-742
[EDT.SRC]WRIEDTMSG.B32;1

Page 20
(6)

: 535 0663 2 PUT_STATUS = \$PUT (RAB = .RAB_ADDR);
: 536 0664 2
: 537 0665 2 IF (NOT .PUT_STATUS) THEN RETURN (.PUT_STATUS);
: 538 0666 2
: 539 0667 2 RETURN (SSS_NORMAL);
: 540 0668 1 END;

! End of routine PRINT

.EXTRN SYSSPUT

| | | | | | | |
|-----------|----|----|-------------------|-------|-----------------|------|
| | | | 0000 00000 PRINT: | .WORD | Save nothing | 0607 |
| | 51 | 04 | AC DD 00002 | MOVL | RAB_ADDR, R1 | 0658 |
| | 50 | 08 | AC DD 00006 | MOVL | TEXT_LINE, R0 | |
| 22 | A1 | 08 | 60 B0 0000A | MOVW | (R0), 34(R1) | |
| 28 | A1 | 04 | A0 DD 0000E | MOVL | 4(R0), 40(R1) | 0659 |
| | | | 51 DD 00013 | PUSHL | R1 | 0663 |
| 00000000G | 00 | | 01 FB 00015 | CALLS | #1, SYSSPUT | |
| | 03 | | 50 E9 0001C | BLBC | PUT_STATUS, 1\$ | 0665 |
| | 50 | | 01 DD 0001F | MOVL | #1, R0 | 0667 |
| | | | 04 00022 1\$: | RET | | 0668 |

: Routine Size: 35 bytes, Routine Base: _EDT\$CODE + 05CC

542 0669 1 ISBTTL 'HEX_TEXT - Return a binary string in hexadecimal'
543 0670 1 ROUTINE HEX_TEXT (! Return a binary string in hex
544 0671 1 OUTPUT DESC, ! Descriptor to receive the hex
545 0672 1 INPUT [EN, ! Number of input bytes
546 0673 1 INPUT_ADDR ! Address of start of input
547 0674 1) =
548
549 0675 1 !++
550 0677 1 FUNCTIONAL DESCRIPTION:
551 0678 1 This routine converts an arbitrary string of bytes into hex, so it
552 can be printed. Early bytes are put to the right of later bytes.
553
554 0682 1 CALLING SEQUENCE:
555
556 0684 1 status.wlc.v = HEX_TEXT (OUTPUT_desc.wt.dx, INPUT_LEN.rl.v, INPUT_ADDR.ra.v)
557
558 0686 1 FORMAL PARAMETERS:
559
560 0687 1 output_desc Where the result text is stored.
561 0689 1 input_len Number of bytes of input
562 0690 1 input_addr Address of first input byte
563
564 0692 1 IMPLICIT INPUTS:
565
566 0693 1 NONE
567
568 0695 1 IMPLICIT OUTPUTS:
569
570 0697 1 NONE
571
572 0698 1 0699 1 COMPLETION STATUS:
573
574 0700 1 0701 1 SSS_NORMAL Normal successful completion
575 0702 1 Any errors from STR\$CONCAT
576 0703 1 Any errors from STR\$COPY_DX
577
578 0704 1
579 0705 1
580 0706 1 SIDE EFFECTS:
581
582 0708 1 Calls STR\$CONCAT and STR\$COPY_DX, thus manipulating string storage.
583
584 0709 1
585 0710 1 --
586
587 0711 1 BEGIN
588
589 0712 2 MAP
590 0713 2 INPUT_ADDR : REF VECTOR [BYTE];
591 0714 2 OUTPUT_DESC : REF BLOCK [8, BYTE];
592
593 0715 2 LOCAL
594 0716 2 INTER_DESC : BLOCK [8, BYTE],
595 0717 2 DIGIT_DESC : BLOCK [8, BYTE],
596 0718 2 DIGIT,
597 0719 2 STATUS;
598 0720 2 INIT_DESCRIPTOR (INTER_DESC);
0721 2 DIGIT_DESC [DSC\$W_LENGTH] = i;

```

599 0726 2      DIGIT_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
600 0727 2      DIGIT_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
601 0728 2      DIGIT_DESC [DSC$A_POINTER] = DIGIT;
602 0729 2
603 0730 2      INCR CHAR_NO FROM 1 TO .INPUT_LEN DO
604 0731 2      BEGIN
605 0732 2
606 0733 2      LOCAL
607 0734 2      CHAR;
608 0735 2
609 0736 2      CHAR = .INPUT_ADDR [.CHAR_NO - 1];
610 0737 2      DIGIT = (.CHAR AND 15) + '0';
611 0738 2
612 0739 2      IF (.DIGIT GTR '9') THEN DIGIT = .DIGIT - 10 - '0' + 'A';
613 0740 2
614 0741 2      STATUS = STR$CONCAT (INTER_DESC, DIGIT_DESC, INTER_DESC);
615 0742 2
616 0743 2      IF (NOT .STATUS) THEN RETURN (.STATUS);
617 0744 2
618 0745 2      DIGIT = (.CHAR^4) + '0';
619 0746 2
620 0747 2      IF (.DIGIT GTR '9') THEN DIGIT = .DIGIT - 10 - '0' + 'A';
621 0748 2
622 0749 2      STATUS = STR$CONCAT (INTER_DESC, DIGIT_DESC, INTER_DESC);
623 0750 2
624 0751 2      IF (NOT .STATUS) THEN RETURN (.STATUS);
625 0752 2
626 0753 2
627 0754 2
628 0755 2      STATUS = STR$COPY DX (.OUTPUT_DESC, INTER_DESC);
629 0756 2      DISCARD_DESCRIPTOR (INTER_DESC);
630 0757 2      RETURN T.STATUS;
631 0758 1      END;

```

! End of routine HEX_TEXT

003C 00000 HEX_TEXT:

| | | | | | | |
|----|----|-------------|-----------------|--------|-------------------------|------|
| | | | | .WORD | Save R2,R3,R4,R5 | 0670 |
| | 55 | 00000000G | 00 9E 00002 | MOVAB | STR\$CONCAT, R5 | |
| | 5E | | 14 C2 00009 | SUBL2 | #20, SP | 0724 |
| | 0C | AE 020E0000 | 8F D0 0000C | MOVL | #34471936, INTER_DESC | |
| | | | 10 AE D4 00014 | CLRL | INTER_DESC+4 | 0725 |
| | 04 | AE 010E0001 | BF D0 00017 | MOVL | #17694721, DIGIT_DESC | |
| | 08 | AE | 6E 9E 0001F | MOVAB | DIGIT, DIGIT_DESC+4 | 0728 |
| | | | 53 D4 00023 | CLRL | CHAR_NO | 0736 |
| | | | 4E 11 00025 | BRB | 48 | |
| | 50 | 53 | OC AC C1 00027 | ADDL3 | INPUT_ADDR, CHAR_NO, R0 | |
| | 52 | FF | 18: A0 9A 0002C | MOVZBL | -1(R0), CHAR | 0737 |
| | 6E | 04 | 00 EF 00030 | EXTZV | #0 #4, CHAR, DIGIT | |
| | | 6E | 30 C0 00035 | ADDL2 | #48 DIGIT | 0739 |
| | | 39 | 6E D1 00038 | CMPL | DIGIT, #57 | |
| | | 6E | 03 15 0003B | BLEQ | 28 | |
| | | 07 | C0 0003D | ADDL2 | #7 DIGIT | 0741 |
| 6E | | OC | AE 9F 00040 | PUSHAB | INTER_DESC | |
| | | 08 | 28: AE 9F 00043 | PUSHAB | DIGIT_DESC | |

EDTSWR1EDTMSG
VO4-000EDTSWR1EDTMSG - write VMSMSG.MSG
HEX_TEXT - Return a binary string in hexadecima16-Sep-1984 02:18:31
14-Sep-1984 12:25:55VAX-11 Bliss-32 v4.0-742
[EDT.SRC]WR1EDTMSG.B32:1Page 23
(7)EXE1
Modu

FPL
VAX1
SYS

| | | | | | | | |
|-----------|----|----|----|-------------|--------|-----------------|------------------------|
| | | 14 | AE | 9F 00046 | PUSHAB | INTER DESC | |
| | | 65 | 03 | FB 00049 | CALLS | #3, STR\$CONCAT | |
| | | 54 | 50 | DD 0004C | MOVL | R0, STATUS | |
| | | 4E | 54 | E9 0004F | BLBC | STATUS, \$8 | |
| | 52 | 52 | 8F | 78 00052 | ASHL | #-4, CHAR, R2 | 0743 |
| | | 6E | 30 | A2 9E 00057 | MOVAB | 48(R2), DIGIT | 0745 |
| | | 39 | 6F | D1 0005B | CMPL | DIGIT, #57 | 0747 |
| | | | 03 | 15 0005E | BLEQ | 35 | |
| | | 6E | 07 | C0 00060 | ADDL2 | #7, DIGIT | |
| | | | 0C | AE 9F 00063 | 38: | PUSHAB | INTER DESC |
| | | | 08 | AE 9F 00066 | | PUSHAB | DIGIT-DESC |
| | | | 14 | AE 9F 00069 | | PUSHAB | INTER DESC |
| | | 65 | 03 | FB 0006C | CALLS | #3, STR\$CONCAT | |
| | | 54 | 50 | DD 0006F | MOVL | R0, STATUS | |
| | | 28 | 54 | E9 00072 | BLBC | STATUS, \$8 | |
| | AD | 53 | 08 | AC F3 00075 | 48: | AOBLEQ | INPUT_LEN, CHAR_NO, 18 |
| | | | 0C | AE 9F 0007A | | PUSHAB | INTER-DESC |
| | | | 04 | AC DD 0007D | | PUSHL | OUTPUT DESC |
| 00000000G | 00 | | 02 | FB 00080 | | CALLS | #2, STR\$COPY_DX |
| | | 54 | 50 | DD 00087 | | MOVL | R0, STATUS |
| 00000000G | 00 | OC | AE | 9F 0008A | | PUSHAB | INTER DESC |
| | | | 01 | FB 0008D | | CALLS | #1, STR\$FREE1 DX |
| | | 09 | 50 | E8 00094 | | BLBS | FREE STATUS, \$8 |
| 00000000G | 00 | | 50 | DD 00097 | | PUSHL | FREE STATUS |
| | | | 01 | FB 00099 | | CALLS | #1, [IB\$STOP |
| | | 50 | 54 | DD 000A0 | 58: | MOVL | STATUS, R0 |
| | | | 04 | 000A3 | | RET | |

; Routine Size: 164 bytes, Routine Base: _EDTSCODE + 05EF

```

633      0759 1 XSBTTL 'PRINTABLE_TEXT - Return a binary string in ASCII, printable'
634      0760 1 ROUTINE PRINTABLE_TEXT (
635      0761 1   OUTPUT DESC,
636      0762 1   INPUT [EN,
637      0763 1   INPUT_ADDR
638      0764 1   ) =
639
640      0765 1 ++
641      0766 1   FUNCTIONAL DESCRIPTION:
642      0767 1
643      0769 1   This routine converts an arbitrary string of bytes into ASCII, representing
644      0770 1   unprintable characters in hexadecimal so the result can be printed.
645
646      0772 1   CALLING SEQUENCE:
647      0774 1   status.wlc.v = PRINTABLE_TEXT (OUTPUT_desc.wt.dx, INPUT_LEN.rl.v, INPUT_ADDR.ra.v)
648
649
650      0776 1   FORMAL PARAMETERS:
651
652      0778 1   OUTPUT DESC      Where the result text is stored.
653      0779 1   INPUT [EN        Number of bytes of input
654      0780 1   INPUT_ADDR     Address of first input byte
655
656      0782 1   IMPLICIT INPUTS:
657      0783 1   NONE
658
659      0785 1   IMPLICIT OUTPUTS:
660      0788 1   NONE
661
662      0789 1   COMPLETION STATUS:
663
664      0791 1   SSS_NORMAL      Normal successful completion
665      0792 1   Any errors from STR$CONCAT
666      0793 1   Any errors from STR$COPY_DX
667
668      0795 1   SIDE EFFECTS:
669      0796 1   Calls STR$CONCAT and STR$COPY_DX, thus manipulating string storage.
670
671      0797 1   --
672      0798 1
673      0799 1
674      0800 1
675      0801 1
676      0802 2   BEGIN
677      0803 2
678      0804 2
679      0805 2   MAP
680      0806 2   INPUT_ADDR : REF VECTOR [ , BYTE],
681      0807 2   OUTPUT_DESC : REF BLOCK [8, BYTE];
682
683      0808 2
684      0809 2   LOCAL
685      0810 2   INTER_DESC : BLOCK [8, BYTE],
686      0811 2   CHAR_DESC : BLOCK [8, BYTE];
687      0812 2   CHARREP : VECTOR [4, BYTE];
688      0813 2   STATUS;
689      0814 2   INIT_DESCRIPTOR (INTER_DESC);
690      0815 2   CHAR_DESC [DSCSW_LENGTH] = 1;

```

690 0816 2 CHAR_DESC [DSCSB_DTYPE] = DSCSK_DTYPE_T;
691 0817 2 CHAR_DESC [DSCSB_CLASS] = DSCSK_CLASS_S;
692 0818 2 CHAR_DESC [DSCSA_POINTER] = CHAR REP [0];
693 0819 2
694 0820 2 INCR CHAR_NO FROM 1 TO .INPUT_LEN DO
695 0821 2 BEGIN
696 0822 2
697 0823 2 LOCAL
698 0824 2 CHAR;
699 0825 2
700 0826 3 CHAR = .INPUT_ADDR [.CHAR_NO - 1];
701 0827 3
702 0828 4 IF ((.CHAR GEQ XX'20') AND
703 0829 4 (.CHAR LSS XX'7F') AND
704 0830 4 (.CHAR NEQ '<') AND
705 0831 4 (.CHAR NEQ '''))
706 0832 3 THEN
707 0833 4 BEGIN
708 0834 4 /* Show character as itself.
709 0835 4 */
710 0836 4
711 0837 4 CHAR REP [0] = .CHAR;
712 0838 4 CHAR_DESC [DSCSW_LENGTH] = 1;
713 0839 4 END
714 0840 3 ELSE
715 0841 4 BEGIN
716 0842 4 /* The character is not printable. Represent it by <>. To avoid
717 0843 4 ambiguity, "<" and ">" are also represented this way. Control characters
718 0844 4 SOH through SUB are represented by <^letter>; others characters by <hex>.
719 0845 4 */
720 0846 4
721 0847 4 CHAR REP [0] = '<';
722 0848 4
723 0849 5 IF (((.CHAR + XX'40') GEQ 'A') AND ((.CHAR + XX'40') LEQ 'Z'))
724 0850 4 THEN
725 0851 5 BEGIN
726 0852 5 CHAR REP [1] = '^';
727 0853 5 CHAR REP [2] = .CHAR + XX'40';
728 0854 5 END
729 0855 4 ELSE
730 0856 5 BEGIN
731 0857 5
732 0858 5 LOCAL
733 0859 5 DIGIT;
734 0860 5
735 0861 5 DIGIT = (.CHAR^4) + '0';
736 0862 5
737 0863 5 IF (.DIGIT GTR '9') THEN DIGIT = .DIGIT - 10 - '0' + 'A';
738 0864 5
739 0865 5 CHAR REP [1] = .DIGIT;
740 0866 5 DIGIT = (.CHAR AND 15) + '0';
741 0867 5
742 0868 5 IF (.DIGIT GTR '9') THEN DIGIT = .DIGIT - 10 - '0' + 'A';
743 0869 5
744 0870 5 CHAR REP [2] = .DIGIT;
745 0871 4
746 0872 4 END;

```

747 0873 4 CHARREP [3] = '>';
748 0874 4 CHARDESC [DSCSW_LENGTH] = 4;
749 0875 4 END;
750 0876 4
751 0877 4 STATUS = STR$CONCAT (INTER_DESC, INTER_DESC, CHAR_DESC);
752 0878 4
753 0879 4 IF ( NOT .STATUS) THEN RETURN (.STATUS);
754 0880 4
755 0881 2 END;
756 0882 2
757 0883 2 STATUS = STR$COPY DX (.OUTPUT_DESC, INTER_DESC);
758 0884 2 DISCARD_DESCRIPTOR (INTER_DESC);
759 0885 2 RETURN T.STATUS);
760 0886 1 END;

```

! End of routine PRINTABLE_TEXT

000C 00000 PRINTABLE_TEXT:

| | | | | | | | |
|----|-------------------|-------------------|--|------|------------|-------------------------|------|
| | | | | | WORD | Save R2,R3 | 0760 |
| 0C | AE 020E0000 | 14 C2 00002 | | | SUBL2 | #20 SP | 0814 |
| | | 8F D0 00005 | | | MOVL | #34471936, INTER_DESC | |
| 04 | AE 010E0001 | 10 AE D4 0000D | | | CLRL | INTER DESC+4 | 0815 |
| 08 | AE | 8F D0 00010 | | | MOVL | #17694721, CHAR_DESC | 0818 |
| | | 6E 9E 00018 | | | MOVAB | CHAR REP, CHAR_DESC+4 | |
| | | 52 D4 0001C | | | CLRL | CHAR_NO | 0826 |
| 50 | | 0090 31 0001E | | 1\$: | BRW | 7\$ | |
| | | 52 AC C1 00021 | | | ADDL3 | INPUT ADDR, CHAR_NO, R0 | |
| | | 51 FF A0 9A 00026 | | | MOVZBL | -1(ROT, CHAR | |
| | | 20 51 D1 0002A | | | CMPL | CHAR, #32 | 0828 |
| | 0000007F | 1C 19 0002D | | | BLSS | 2\$ | 0829 |
| | | 51 D1 0002F | | | CMPL | CHAR, #127 | |
| | | 13 18 00036 | | | BGEQ | 2\$ | |
| | | 3C 51 D1 00038 | | | BEQL | CHAR, #60 | 0830 |
| | | 0E 13 0003B | | | CMPL | CHAR, #34 | |
| | | 22 51 D1 0003D | | | BEQL | 2\$ | 0831 |
| | | 09 13 00040 | | | MOVW | CHAR, CHAR REP | 0837 |
| | | 51 90 00042 | | | MOVW | #1, CHAR_DESC | 0838 |
| 04 | AE | 01 B0 00045 | | | BRB | 6\$ | 0828 |
| | | 50 11 00049 | | | MOVAB | #60, CHAR REP | 0847 |
| | | 6E 3C 90 0004B | | 2\$: | 64(R1), R0 | | 0849 |
| | 00000041 | 50 8F A1 9E 0004E | | | CMPL | R0, #65 | |
| | | 50 D1 00052 | | | BLSS | 3\$ | |
| | | 10 19 00059 | | | CMPL | R0, #90 | |
| | 0000005A | 8F 50 D1 0005B | | | BGTR | 3\$ | |
| | | 07 14 00062 | | | MOVW | #94, CHAR REP+1 | 0852 |
| | | 8F 90 00064 | | | MOVW | | 0853 |
| 01 | AE | 5E 24 11 00069 | | | BRB | 5\$ | |
| 50 | | 51 FC 8F 78 0006B | | 3\$: | ASHL | #-4, CHAR, R0 | 0861 |
| | | 50 30 C0 00070 | | | ADDL2 | #48, DIGIT | |
| | | 39 50 D1 00073 | | | CMPL | DIGIT, #57 | 0863 |
| | | 03 15 00076 | | | BLEQ | 4\$ | |
| | | 07 C0 00078 | | | ADDL2 | #7, DIGIT | |
| 50 | 01 AE 50 90 0007B | 50 EF 0007F | | 4\$: | MOVW | DIGIT, CHAR REP+1 | 0865 |
| | | 30 C0 00084 | | | EXTZV | #0, #4, CHAR, DIGIT | |
| | | | | | ADDL2 | #48, DIGIT | 0866 |

EDT\$WRIEDTMSG
V04-000

EDT\$WRIEDTMSG - write VMSMSG.MSG
PRINTABLE_TEXT - Return a binary string in ASCII

N 4

16-Sep-1984 02:18:31
14-Sep-1984 12:25:55

VAX-11 Bliss-32 V4.0-742
[EDT.SRC]WRIEDTMSG.B32;1

Page 27
(B)

\$21

| | | | | | | | |
|------|----|-----------|----|----------------|--------|-----------------------------|------------------|
| | | 39 | 50 | D1 00087 | CMPL | DIGIT, #57 | 0868 |
| | | 50 | 03 | 15 0008A | BLEQ | 5\$ | |
| | 02 | AE | 07 | C0 0008C | ADDL2 | #7, DIGIT | |
| | 03 | AE | 50 | 90 0008F | 5\$: | MOV B | CHAR, CHAR REP+2 |
| | 04 | AE | 3E | 90 00093 | MOV B | #62, CHAR REP+3 | |
| | | | 04 | B0 00097 | MOV W | #4, CHAR DESC | |
| | | | 10 | AE 9F 0009E | PUSHAB | CHAR DESC | |
| | | | 14 | AE 9F 000A1 | PUSHAB | INTER-DESC | |
| | | 00000000G | 00 | 03 FB 000A4 | CALLS | INTER-DESC | |
| | | | 53 | 50 D0 000AB | MOVL | #3, STR\$CONCAT | |
| | | | 2D | 53 E9 000AE | BLBC | RO, STATUS | |
| FF69 | 52 | | 01 | 08 AC F1 000B1 | 7\$: | STATUS, 8\$ | 0879 |
| | | | | 0C AE 9F 000B8 | ACBL | INPUT_LEN, #1, CHAR_NO, 1\$ | 0820 |
| | | | | 04 AC DD 000BB | PUSHAB | INTER-DESC | 4851 |
| | | 00000000G | 00 | 02 FB 000BE | PUSHL | OUTPUT DESC | 7FFE |
| | | | 53 | 50 D0 000C5 | CALLS | #2, STR\$COPY_DX | 8000 |
| | | 00000000G | 00 | 0C AE 9F 000C8 | MOVL | RO, STATUS | 8000 |
| | | | 09 | 01 FB 000CB | PUSHAB | INTER DESC | 8000 |
| | | 00000000G | 00 | 50 E8 000D2 | CALLS | #1, STR\$FREE1 DX | 8000 |
| | | | 50 | 50 DD 000D5 | BLBS | FREE_STATUS, 8\$ | 8000 |
| | | 00000000G | 00 | 01 FB 000D7 | PUSHL | FREE_STATUS | 8000 |
| | | | 50 | 53 D0 000DE | CALLS | #1, [IB\$STOP | 8001 |
| | | | | 04 000E1 | RET | STATUS, RO | 0885 |
| | | | | | MOVL | | 0886 |

; Routine Size: 226 bytes, Routine Base: _EDT\$CODE + 0693

; 761 0887 1 !<BLF/PAGE>

EDT\$WRIEDTMSG 16-Sep-1984 02:18:31 VAX-11 Bliss-32 V4.0-742
V04-000 PRINTABLE_TEXT - Return a binary string in ASCII 14-Sep-1984 12:25:55 [EDT.SRC]WRIEDTMSG.B32;1 Page 28 (9) -\$2

: 763 0888 1 END
: 764 0889 1
: 765 0890 0 ELUDOM ! End of module EDT\$WRIEDTMSG

.EXTRN LIB\$STOP

PSECT SUMMARY

| Name | Bytes | Attributes |
|------------|-------|---|
| _EDT\$CODE | 1909 | NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2) |

Library Statistics

| File | Total | Symbols Loaded | Percent | Pages Mapped | Processing Time |
|---------------------------------------|-------|-------------------|---------|-----------------|--------------------|
| \$_\$255\$DUA28:[SYSLIB]STARLET.L32;1 | 9776 | 106 | 1 | 581 | 00:02.4 |

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LISS:WRIEDTMSG/OBJ=OBJ\$:WRIEDTMSG MSRC\$:WRIEDTMSG.B32/UPDATE=(ENH\$:W
: RIEDTMSG)

: Size: 1601 code + 308 data bytes
: Run Time: 01:14.7
: Elapsed Time: 02:10.7
: Lines/CPU Min: 714
: Lexemes/CPU-Min: 11820
: Memory Used: 334 pages
: Compilation Complete

0142 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

